CLAIM AMENDMENTS

Please amend the claims by amending claim 6, and adding new claims 38-42, all without prejudice, as indicated on the following listing of all the claims in the present application after this Amendment:

1. (original) A method of storing data in a memory array that includes a plurality of individually erasable erase blocks arranged in separately programmable planes, comprising:

selecting a number of erase blocks from different planes of the array to form an adaptive metablock, the number of erase blocks in the adaptive metablock selected from a range of possible values; and

programming the erase blocks in the adaptive metablock in parallel.

2. (original) The method of claim 1 wherein the range of possible values is from one to the number of separately programmable planes in the memory array.

3. (original) The method of claim 1 wherein programming the erase blocks in the adaptive metablock in parallel takes place in parallel with programming a plurality of erase blocks that are not in the adaptive metablock.

4. (original) The method of claim 3 wherein the plurality of erase blocks that are not in the adaptive metablock are in another adaptive metablock.

5. (original) The method of claim 1 wherein selecting a number of erase blocks comprises selecting a number of planes, an individual plane being chosen according to the number of available erase blocks in the plane.

6. (currently amended) The method of claim 1 wherein the data to be stored comprises addressable units of data, logical groups are individually formed from a number of logically sequential addressable units of data that is equal to the number of addressable units of data that

may be stored in an erase block, a logical group is programmed to an adaptive metablock such that each of the erase blocks in the adaptive metablock contains at least one addressable unit of data from the logical group.

7. (original) A method of storing data in a memory array that includes erase blocks arranged in separately programmable planes, comprising:

selecting a plurality of planes of the memory array, a plane of the plurality being individually selected by an algorithm based on the availability of erase blocks in the plane;

forming an adaptive metablock from a plurality of erase blocks, each of the plurality of erase blocks being from a different one of the selected planes; and

programming the erase blocks of the adaptive metablock in parallel.

8. (original) The method of claim 7 wherein erase blocks from different planes of the array are not located at the same position within their respective planes.

9. (original) The method of claim 7 wherein the number of erase blocks is selected from a range of possible values.

10. (original) The method of claim 7 further comprising forming a second adaptive metablock and programming the adaptive metablock and the second adaptive metablock in parallel.

11. (original) The method of claim 10 wherein the number of erase blocks in the adaptive metablock and the number of erase blocks in the second adaptive metablock are not equal.

12. (original) A method of storing data in a memory array that includes a plurality of multi-sector erase blocks arranged in separately programmable planes, comprising:

forming a plurality of logical groups from data to be stored in the array, a logical group comprising a number of sectors of data that is equal to the number of sectors of data stored in one erase block of the memory array;

forming an adaptive logical block from the plurality of logical groups;

forming an adaptive metablock from the plurality of multi-sector erase blocks;

storing the adaptive logical block in the adaptive metablock of the memory array such that an individual one of the plurality of multi-sector erase block contains portions of each of the plurality of logical groups; and

individually recording address information for the logical groups of the adaptive logical block.

13. (original) The method of claim 12 wherein the adaptive logical block is stored in the adaptive metablock of the memory array by programming the plurality of multi-sector erase blocks of the memory array in parallel.

14. (original) The method of claim 12 wherein the number of logical groups within the adaptive logical block is individually determined for the adaptive logical block.

15. (original) A non-volatile memory system, comprising:

an array of memory cells arranged in separately programmable planes;

a plane having multiple erase blocks, an erase block being the smallest unit of the array that may be individually erased;

the system configured to select a number of planes according to characteristics of data to be stored; and

the system configured to program individual erase blocks within selected planes in parallel.

16. (original) The system of claim 15 wherein an individual plane is selected according to the number of available erase blocks remaining in the individual plane.

17. (original) The system of claim 15 further comprising a second array of memory cells;

wherein the first array of memory cells is in a flash memory and the second array of memory cells is in a non-volatile random access memory.

18. (original) A method of updating data within a non-volatile memory array having a plurality of individually erasable blocks configured to form a first adaptive metablock, comprising;

receiving updated data from a host, the updated data to replace data stored within the adaptive metablock;

storing the updated data in a second adaptive metablock comprised only of erase blocks that contain updated data; and

storing original data from the first adaptive metablock in a third adaptive metablock comprised only of erase blocks that contain no updated data.

19. (original) The method of claim 18 wherein the second adaptive metablock and the third adaptive metablock are both smaller than the first adaptive metablock.

20. (original) The method of claim 18 wherein the first adaptive metablock has a first number of erase blocks, the second adaptive metablock has a second number of erase blocks and the third adaptive metablock has a third number of erase blocks; and

the sum of the second number and the third number is equal to the first number.

21. (original) A method of updating data stored in a non-volatile memory array having a first plurality of original adaptive metablocks, an individual original adaptive metablock storing an original adaptive logical block having a plurality of original logical groups of data, comprising:

receiving updated data to replace old data;

forming updated logical groups, an updated logical group comprising sequential data having the same logical address range as an original logical group, at least some of the data in an updated logical group being updated data;

forming updated adaptive logical blocks, an updated adaptive logical block being comprised of a number of updated logical groups, the number of updated logical groups in an updated adaptive logical block being independent of the number of original logical groups in an original adaptive logical block; and

programming an updated adaptive logical block to an updated adaptive metablock.

22. (original) The method of claim 21 wherein the number of updated logical groups in an updated adaptive logical block is less than the number of original logical groups in an original adaptive logical block.

23. (original) The method of claim 21 wherein updated logical groups in an updated adaptive logical block correspond to original logical groups in more than one original adaptive logical block.

24. (original) A method of managing erased blocks of a non-volatile memory array, an erased block being an erase block that is in condition for programming, an erase block being the minimum unit of erase of the non-volatile array, comprising:

maintaining a first list in a volatile memory that indicates a first plurality of erased blocks from which individual erased blocks are directly selected for programming; and

updating the first list to delete an entry for an erased block that is selected for programming thus producing an updated first list; and

copying the contents of the updated first list to a second list in the non-volatile memory in response to updating the first list.

25. (original) The method of claim 24 wherein the second list is programmed to the non-volatile memory as part of a first addressable data packet.

26. (original) The method of claim 25 wherein the second list is updated by programming a second addressable data packet to the non-volatile memory and indicating that the first addressable data packet is obsolete.

27. (original) The method of claim 24 further comprising maintaining a third list that indicates a second plurality of erased blocks; and

exchanging individual ones of the first plurality of erased blocks and individual ones of the second plurality of erased blocks between the first list and the third list.

28. (original) The method of claim 27 wherein the rate of exchanging is selected according to criteria including wear leveling.

29. (original) The method of claim 27 further comprising maintaining a fourth list that indicates a third plurality of erased blocks and exchanging individual ones of the third plurality of erased blocks and individual ones of the second plurality of erased blocks between the third list and the fourth list.

30. (original) The method of claim 29 wherein the non-volatile memory array has a plurality of planes, individual ones of the first and second plurality of erase blocks being from any of the plurality of planes, individual ones of the third plurality of erase blocks being from a first plane of the array.

31. (original) A method of storing a plurality of logical groups of data in a memory array, a logical group containing data that is equal to the amount of data stored in an erase block of the memory array, comprising:

forming a metablock containing a plurality of erase blocks;

storing the plurality of logical groups including a first and a second logical group, such that each of the plurality of erase blocks of the metablock contains a portion of the first logical group and a portion of the second logical group; and

maintaining a table having a first entry for the first logical group and a second entry for the second logical group, the first entry including the physical location of a first one of the plurality of erase blocks and the second entry containing the physical location of a second of the plurality of erase blocks.

32. (original) The method of claim 31 wherein the first entry includes the number of erase blocks in the metablock and the logical position of the first logical group within the plurality of logical groups.

33. (original) The method of claim 31 wherein each one of the plurality of logical groups has an entry in the table, the number of logical groups in the plurality of logical groups is equal to the number of erase blocks in the plurality of erase blocks and the location of each one of the plurality of erase blocks is given by an entry in the table.

34. (original) A method of storing a plurality of first addressable units of data in a non-volatile memory array that has a minimum unit of erase of an erase block, an erase block containing an integer number of addressable units of data, comprising:

forming a second addressable unit of data from the integer number of first addressable units of data;

writing the second addressable unit of data to the non-volatile memory array such that a first portion of the second addressable unit of data is stored in a first erase block and a second portion of the second addressable unit of data is stored in a second erase block; and

individually recording location information for the second addressable unit of data that may be used to determine the physical location of the second addressable unit of data in the non-volatile memory array.

35. (original) The method of claim 34 wherein the first addressable unit of data is a sector and the second addressable unit of data is a logical group.

36. (original) The method of claim 34 wherein the physical location of the second addressable unit of data is determined using the location information for the second addressable unit of data and from location information for other addressable units of data.

37. (original) A method of storing data in a memory array that includes a plurality of multi-sector erase blocks arranged in separately programmable planes, an erase block being the minimum unit of erase of the memory array, comprising:

forming a plurality of logical groups from data to be stored in the array, a logical group comprising a number of sectors of data that is equal to the number of sectors of data stored in one erase block of the memory array;

forming an adaptive metablock from a number of erase blocks, the number selected from a range of numbers;

storing a logical group in the adaptive metablock of the memory array such that each of the number of erase blocks of the adaptive metablock contains a portion of the logical group.

38. (New) A method of storing data in a nonvolatile memory connected to a host, comprising:

receiving a first number of logically sequential sectors of data from the host;

selecting a second number of erase blocks for storage of the first number of sectors, each of the second number of erase blocks located in a different plane, the second number being less than the number of planes in the memory, the second number being the smallest number of erase blocks that can contain the first number of sectors; and

programming the first number of sectors to the second number of erase blocks in parallel.

39. (New) The method of claim 38 wherein ones of the first number of sectors are stored in an individual one of the second number of erase blocks in a non-sequential order.

40. (New) The method of claim 38 further comprising, in parallel with programming the first number of sectors to the second number of erase blocks, programming additional sectors to additional erase blocks, the additional sectors having a logical address range that is separated from the logical address range of the first number of sectors.

41. (New) The method of claim 40 further comprising subsequently erasing the second number of erase blocks in parallel without erasing the additional erase blocks.

42. (New) The method of claim 41 further comprising subsequently selecting a first at least one of the second number of erase blocks for programming without selecting a second at least one of the second number of erase blocks for programming.